# Software Requirements

Version 4, last updated by Zac and Carol at 2021-4-13

# Software Requirements Specification (SRS)

Revision History:

|  |  |  |
| --- | --- | --- |
| Date | Author | Description |
| 4-12-2021 | Carol Liao | First Version |
| 4-13-2021 | Zac. Chen | * Add some content into background part. * Rewrite several paragraphs to make it more fluent to read. * Fixed the inconsistencies of formats. * Update and adjust the catalog. |
| 4-13-2021 | A. Mota | Proofread several grammatical errors in documents. |
| 4-13-2021 | Zac. & Carol | * Adjust the user requirements to avoid their duplication between each other. * Some academic terms have been replaced to make it easier to understand. |
| 4-24-2021 | Zac. & Carol&Doris | Modify and adjust the user case part and fill in the form of table. |
| 4-24-2021 | A. Mota | Add the use cases diagram. |

Contents

[Software Requirements 1](#_Toc69735911)

[Software Requirements Specification (SRS) 1](#_Toc69735912)

[1.  Introduction 3](#_Toc69735913)

[1.1    Intended Audience and Purpose 3](#_Toc69735914)

[1.2    How to use this documentation 3](#_Toc69735915)

[2.  Concept of Operations 4](#_Toc69735916)

[2.1    System Context 5](#_Toc69735917)

[2.2 System capabilities 5](#_Toc69735918)

[3.  Use Cases 5](#_Toc69735919)

[Case 1: The user wants to get all the key points of the bone 5](#_Toc69735920)

[Case 2: The user wants to get the points of the two bones that form the Cobb 6](#_Toc69735921)

[Case 3: The user wants to get the Cobb value. 6](#_Toc69735922)

[4.    Behavioral Requirements 6](#_Toc69735923)

[4.1 System Inputs and Outputs 6](#_Toc69735924)

[5.  Quality Requirements 7](#_Toc69735925)

[6.    Expected Subsets 7](#_Toc69735926)

[7.    Fundamental Assumptions 8](#_Toc69735927)

[8.    Expected Changes 8](#_Toc69735928)

[9.    Appendices 8](#_Toc69735929)

[9.1    Definitions and acronyms 8](#_Toc69735930)

[9.2    References 9](#_Toc69735931)

## 1.  Introduction

### 1.1    Intended Audience and Purpose

This document is intended to provide information guiding the installation and development process, ensuring that all system requirements are met. The following entities may find the document useful:

* Primary Customer - This page will detail all the application requirements as understood by the production team. The customer should be able to determine that their requirements will be correctly reflected in the final product through the information found on this page.
* User - A prospective user will be able to use this document to identify the main functionality included in the application. Furthermore, the application will have a set of system requirements before the application can be ran. Details regarding these requirements can be found here.
* Development Team - Details of specific requirements that the final software build must include will be located here. Developers can use this document to ensure the software addresses each of these requirements.
* QA Team - By developing testing procedures founded in the system requirements, the QA Team can create a comprehensive testing regimen that will guarantee requirements are met.

### 1.2    How to use this documentation

Description of Contents:  
  
1. Introduction – How does this document fit in the different entities.

2. Concept of Operations - Describes the faced problem and the solution to be implemented.  
 2.1 System Context - Details any specific system requirements the application will require to run for diverse operating systems.  
 2.2 System Capabilities – Overview of all capabilities available to the user in the address book.  
  2.3 Use cases - A detailed look at each functional requirement, describing the application context both before and after an action is taken.

3. Behavioral Requirements - How will the application interact with a user.  
  3.1 Input and output requirements – A description of allowed inputs and generated outputs.  
    3.1.1 Input - Describes any restrictions that will be placed on allowed input.  
    3.1.2 Output - Describes the range of outputs that can be generated.  
  3.2 Detailed Output Behavior - Output descriptions in prose.

4. Quality Requirements - Requirements not pertaining to the function of the application will be listed here.

5. Expected Subsets - Expected levels of functionality at checkpoints during development.

6. Fundamental Assumptions - Some specifics about input, output, or behavior upon which other requirements are founded will be listed here.

7. Expected Changes - Future features and directions the project is expected to take.

8. Appendices - Details aiding the understanding of this document  
  8.1 Definitions and acronyms - Any technical terms or abbreviations will be spelled out here for ease while using the document  
    8.1.1 Definitions - Definitions of technical or unusual terminology.  
    8.1.2 Acronyms and Abbreviations - Any abbreviated terms will be expanded here.  
  8.2 References - any external references necessary or helpful to understanding this document will be listed here.

## 2.  Concept of Operations

Scoliosis is a common disease in the growth and development stage of adolescents, which is difficult to be detected in the early stage of onset. The incidence of adolescent idiopathic scoliosis in China is about 0.5%-3%. Teenagers are in a critical period of physical growth and development. Due to the lack of exercise, learning and incorrect walking posture and other factors, spinal lesions will occur. If not treated in time, it will bring irreversible damage to the patient's spine and even seriously influence of the patient's physical form, mental health. In some cases, it may cause life risk. Usually, the orthopedic surgeon manually measures and calculates the Cobb Angle to determine the severity of scoliosis based on the shape of the spine shown in X-ray images. This way of diagnosis inevitably increases the workload of doctors. Apart from that, the standards between doctors are different. With the continuous development of computer artificial intelligence technology, computer aided diagnosis has become an important way to assist doctors in diagnosis, and has achieved significant results, which provides new inspirations for the diagnosis of scoliosis.

Our purpose is to create an efficient and accurate deep learning algorithm to assist in the diagnosis of scoliosis.

The developer needs to build relative environment first. During the process, spinal X-ray images are passed in as input of the algorithm. The algorithm will return the results of Cobb Angle and label of the image as output.

### 2.1    System Context

**System Requirements:**

**Support Windows/Mac OS X/Linux，The following runtime environments need to be installed and configured in advance。**

Windows:

Windows 10/8/7, Python 3.0+, Pytorch1.1.0+, OpenCV-Python

Mac OS X:   
Mac OS X 10+, Python 3.0+, Pytorch1.1.0+, OpenCV-Python

Linux:

Linux 3+, Python 3.0+, Pytorch11.0+, OpenCV-Python

### 

### 2.2 System capabilities

After installing the runtime environment, developer users can call this algorithm program in their own programs. This algorithm program can be based on the spine X-ray provided by the user and related command options, feedback to the user calculation results.

## 3.  Use Cases

### Figure 1: Use case diagram

### Case 1: The user wants to get all the key points of the bone

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | The user wants to get all the key points of the bone | | |
| Version | 6 | Created (date): | 2021-4-24 |
| Author | *Zachary Chen & Carol Liao & Doris Wang* | | |
| Source | \ | | |
| Goals | Get all the key points of the bone | | |
| Summary | \ | | |
| Actors | *Server* | | |
| Trigger | \ | | |
| Precondition | The spinal X-ray image has been stored on the host server. | | |
| Frequency | Once every time you use the application | | |
| Postconditions | Server get the status and points | | |
| Diagram | \ | | |

|  |  |  |
| --- | --- | --- |
| **Basic Flow** | *Actor* | *System* |
|  | The server developer calls this algorithm program by passing in the path address of the given spinal X-ray image and command options (-keypoint) | Algorithm program receive the input. |
|  |  | Algorithm program processes the image and calculates the output. |
|  |  | The algorithm returns a JSON string, with the "status" key corresponding to "success" and the "points" key contains a Json Array, which each terms contains 4 x value and 4 y value for each bone.  e.g. {  “status”: “success”,  “points”: [  {  “1”: [x1,y1],  “2”:[x2,y2],  “3”:[x3,y3]  },  {  ……  }  ]  } |

|  |  |  |
| --- | --- | --- |
| **Alternative Flow** | *Actor* | *System* |
|  | The server developer calls this algorithm program by passing in the path address of the given spinal X-ray image and command options (-keypoint) | Algorithm program receive the input. |
|  |  | The source image path does not have read permission or the destination image path does not have write permission. |
|  |  | The "status" key in the returned JSON string corresponds to "failed". |

### Case 2: The user wants to get the points of the bones that form the Cobb

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | The user wants to get the points of the bones that form the Cobb | | |
| Version | 6 | Created (date): | 2021-4-24 |
| Author | *Zachary Chen & Carol Liao & Doris Wang* | | |
| Source | \ | | |
| Goals | Get all the points of the two that form the Cobb | | |
| Summary | \ | | |
| Actors | *Server* | | |
| Trigger | \ | | |
| Precondition | The spinal X-ray image has been stored on the host server. | | |
| Frequency | Once every time you use the application | | |
| Postconditions | Server get status and two terms of points for each bone | | |
| Diagram | \ | | |

|  |  |  |
| --- | --- | --- |
| **Basic Flow** | *Actor* | *System* |
|  | The server developer calls this algorithm program by passing in the path address of the given spinal X-ray image and command options (-locate) | Algorithm program receive the input. |
|  |  | Algorithm program processes the image and calculates the output. |
|  |  | The algorithm returns a JSON string, with the "status" key corresponding to "success" and “points” key contains a JsonArray, which contains two terms and each term contains 4 x value and 4 y value for each bone.  e.g. {  “status”: “success”,  “points”: {{‘1’: [[x1,y1],[x2,y2],[x3,y3],[x4,y4]]}  , {‘2’: [[x1,y1],[x2,y2],[x3,y3],[x4,y4]]}  ,{ ‘3’: [[x1,y1],[x2,y2],[x3,y3],[x4,y4]]}  } } //each cobb have a list of points(order:top-right,top-left,bottom-right,bottom-left) |

|  |  |  |
| --- | --- | --- |
| **Alternative Flow** | *Actor* | *System* |
|  | The server developer calls this algorithm program by passing in the path address of the given spinal X-ray image and command options (-locate) | Algorithm program receive the input. |
|  |  | The source image path does not have read permission or the destination image path does not have write permission. |
|  |  | The "status" key in the returned JSON string corresponds to "failed". |

### Case 3: The user wants to get the Cobb value.

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | The user wants to get the Cobb value | | |
| Version | 6 | Created (date): | 2021-4-24 |
| Author | *Zachary Chen & Carol Liao* | | |
| Source | \ | | |
| Goals | Get the Cobb value | | |
| Summary | \ | | |
| Actors | *Server* | | |
| Trigger | \ | | |
| Precondition | The spinal X-ray image has been stored on the host server. | | |
| Frequency | Once every time you use the application | | |
| Postconditions | Server get status and cobb value | | |
| Diagram | \ | | |

|  |  |  |
| --- | --- | --- |
| **Basic Flow** | *Actor* | *System* |
|  | The server developer calls this algorithm program by passing in the path address of the given spinal X-ray image and command options (-cobb) | Algorithm program receive the input. |
|  |  | Algorithm program processes the image and calculates the output. |
|  |  | The algorithm returns a JSON string, with the "status" key corresponding to "success" and the “cobb\_value” key corresponding to the value of cobb. |

|  |  |  |
| --- | --- | --- |
| **Alternative Flow** | *Actor* | *System* |
|  | The server developer calls this algorithm program by passing in the path address of the given spinal X-ray image and command options (-cobb) | Algorithm program receive the input. |
|  |  | The source image path does not have read permission or the destination image path does not have write permission. |
|  |  | The "status" key in the returned JSON string corresponds to "failed". |

## 4.    Behavioral Requirements

### 4.1 System Inputs and Outputs

#### 4.1.1 Inputs

The input comes from a call from the developer applying the algorithm.

Inputs When Call This Algorithm  
   \*Path: the path of the computed spinal X-ray image.

   \*Command: Selecting the option on the spinal X-ray image to be computed.

-keypoint：Obtaining the labelling of key points for obtaining all bone masses for a given spinal X-ray.

-locate：Obtaining the key points of all the bone fragments and marking the position of the bone fragments constituting Cobb Angle.

-cobb：Obtaining the key points of all the bone fragments, marking the position of the bone fragments constituting Cobb Angle and Cobb value.

#### 4.1.2 Outputs

The output is the output of the result of running this algorithm

Outputs When Call This Algorithm

\*json: The return value is rendered as a JSON string. The JSON string has the following keys

“status”: Whether the algorithm runs successfully.

“path”: The storage address of the marked image.

“cobb”: The Cobb Angle value。

## 5.  Quality Requirements

The application must be competitive with similar applications in regards to performance, reliability and scalability.

Performance：Responsiveness to user’s response time

\* Algorithm execution time should not exceed 500ms.

\* When an error returns, the response time should not exceed 200ms

Reliability：Affecting the validity of the output results

\*The error of the algorithm should be within 20%

Scalability: Ease of extending application capabilities

\* Application should be modularized such that adding/extending features and functions only require changes to a single component and the interface with that component, if applicable.

## 6.    Expected Subsets

L0:  
Cobb Angle can be returned according to the user's input spinal X-ray image and related results can be labeled on the spinal X-ray image.  
  
L1:  
Users can input the marked images to retrain the algorithm model.

## 7.    Fundamental Assumptions

\* The algorithm doesn't get stuck because of the specificity of the images provided by the user.

\* The accuracy of the algorithm can reach 85%

## 8.    Expected Changes

  To further improve accuracy, making the algorithm more reliable and faster.

## 9.    Appendices

### 9.1    Definitions and acronyms

#### 9.1.1    Definitions

|  |  |
| --- | --- |
| **Keyword** | **Definitions** |
| Cobb | The greatest angle at a particular region of the vertebral column, when measured from the superior endplate of a superior vertebra to the inferior endplate of an inferior vertebra. |
|  |  |
|  |  |
|  |  |

#### 9.1.2    Acronyms and abbreviations

|  |  |
| --- | --- |
| **Acronym or**  **Abbreviation** | **Definitions** |
| QA | Quality Assurance |
| JSON | JavaScript Object Notation |
|  |  |

### 9.2    References

Comments are disabled for this space. In order to enable comments, Messages tool must be added to project.

You can add Messages tool from Tools section on the Admin tab.